Multi Agent Systems Project: Traffic Flow Optimization with Intelligent Traffic-Lights Agents

Marco Bianchi, Emanuele Trossarello, Alessandro Zonato

June 14, 2025

Abstract

In this report we present the design and implementation of a multi-agent system for traffic flow optimization in the urban city of Turin, Italy. Using the GAMA platform, we implemented a network of intelligent traffic lights capable of inter-agent communication based on the FIPA protocol, following an ask-and-answer methodology. As a primary objective we intended to improve urban mobility by dynamically adapting traffic light behavior to current road and intersections conditions. As a proxy for fuel consumption and emission production, we measured the mean positive acceleration of vehicles across various traffic scenarios. The simulation results show that the proposed system effectively enhances traffic flow and leads to a measurable decrease in vehicle acceleration, indicating less emissions and potential environmental benefits.

1 Introduction

- Context and Motivation - Urban traffic congestion represents a critical challenge in modern cities like Turin, leading to concerning problems such as traffic jams blocking the viability, elevated fuel consumption due to unnecessary accelerations, and higher pollutant emissions. Efficient traffic management is therefore essential to improve mobility and reduce environmental impact. Traditional traffic light systems often operate on fixed timing schedules: they lack adaptability to dynamic traffic conditions, which limits their effectiveness in mitigating congestion. Therefore, a data-driven and intelligent approach to traffic management has become essential for modern urban planning and transportation engineering.

- Recent Work - The application of multi-agent systems to urban traffic control has been an active area of research in recent years. Traditional traffic management approaches often rely on fixed-time or actuated signal control, which lack the flexibility to respond dynamically to fluctuating traffic demands. Common real-world implementations typically use sensors placed near intersections -such as inductive loops or cameras— to detect vehicle presence and adjust signal timing accordingly. While these sensor-based systems provide some level of reactivity, they often operate in isolation without broader coordination, limiting the effect in managing traffic across larger urban networks.

Another widely used strategy is the implementation of "green waves," where traffic lights are synchronized along main corridors to create continuous flows of vehicles. Although green waves can reduce stops and delays under certain conditions, they are generally designed based on historical traffic patterns and do not adapt well to real-time variations or unpredictable congestion. This lack of real-time adaptability can result in suboptimal traffic flow when conditions deviate from expected patterns.

We can also find some recent simulation-based works in literature that paved the way to a MAS approach to traffic flow optimization, such as [1], [6], [9], and [10].

- Our Contribution - We present a decentralized multi-agent system for traffic control developed for the city of Turin, Italy. We used the GAMA platform [3], [7] to model traffic lights as independent agents that make decisions and coordinate with each other in real time. The main innovation of our work is the use of the FIPA standard for communication between traffic light agents. Through an ask-and-answer protocol, agents exchange informations about local traffic conditions in their neighborhood and adjust their signal timing cooperatively, to improve overall traffic flow.

Unlike many existing systems that rely on fixed timing or simplified communication methods, our approach allows for flexible and standardized interaction, making the system more scalable and adaptable to changing traffic patterns. Additionally, we use the (positive) cumulative acceleration of vehicles as a proxy to estimate fuel consumption and emissions. This metric captures the effects of acceleration events, which are closely related to increased fuel use and pollutant output, as we can find in [4], [5]. Simulation results demonstrate that this communication-driven coordination can effectively improve emergency situations and reduce unnecessary vehicle acceleration, highlighting the potential environmental benefits of the proposed system. However there is a small trade-off to pay in terms of mean speed and percentage of stopped vehicles, due to the fact that traffic lights need to interrogate one another in order to request green light. Yet it is a promising approach which needs to be examined in depth in the near future.

The following sections of this report describe the system architecture and communication protocol - with some details of the code explained -, the simulation setup and the experimental methodology. Then we present the analysis and results, conclusions, and possible future developments.

2 Model Structure

This section outlines the core structure of our model, focusing on how the urban environment is represented and how agents are embodied into it. The model is built around a graph-based abstraction of a portion of the city of Turin. This structure supports a flexible and scalable simulation framework. We also introduce different species of agents used in the model -vehicles, buses, and traffic lights- and describe their roles within the system. This setup establishes the framework in which our control strategies are implemented and tested.

2.1 The Graph

To construct the spatial layout, we used QGIS [2] to extract a portion of the city map, [1] and imported it into the GAMA platform. The resulting map was converted into a **weighted-directed graph**, where intersections are represented as nodes and they are controlled by traffic-lights agents. Roads connecting them are modeled as directed and weighted edges.

Each road segment in the graph is categorized based on the following criteria: a priority level, the direction relative to the intersection, and an attribute of parity for counting them. Regarding the **priority**, roads are divided into *primary, secondary, tertiary* to underline their importance in terms of mobility and dimensions. Additionally, roads are labeled as either *incoming* or *outgoing* with respect to a given intersection. This **direction** feature helps traffic-light agents to determine which segments to monitor and to manage during signal updates. Finally roads have a **parity** property, and they are categorized as either *even* or *odd*, a convention used to simulate right-of-way dynamics in situations where multiple roads converge.

2.2 Traffic-Lights

Traffic-light agents are placed at each node that satisfies specific criteria[2] on its incoming and outgoing roads. These intersections are able to operate in three distinct scenarios:

- **Fixed timing**: a basic, non-adaptive mode where traffic-lights follow a fixed time cycle, simulating a traditional timer-based control system. This is the most common system used in real cities and it doesn't account for flexibility or changes in traffic flow.
- **Reactive**: agents collect informations from nearby incoming and outgoing roads (e.g., vehicle count, density) in a locally adaptive scenario and adjusts the light phases accordingly, without any communication with neighboring intersections. Despite its simplicity, this method boosts performance and traffic-flow optimization.
- **Flow response communication**: an advanced mode for traffic-lights agents to actively communicate with adjacent intersections using the FIPA protocol. What happens is an exchange of structured messages in an ask-and-answer format to coordinate decisions and optimize traffic flow cooperatively. The aim is to decrease the (positive) cumulative acceleration and to increase the mean flux of vehicles per hour.

2.3 Other Vehicles

In addition to cars, we included a public transportation infrastructure. Bus lines are statically defined within the network, and buses are represented as distinct agents belonging to the **vehicle** class. They follow predefined routes and they are designed to reflect some realistic constraints concerning public transports, including route fidelity and priority at intersections. To additionally represent priority over other vehicles, buses are also given a considerable weight in right-of-way determination logic.

[1] We used a map of approximately $35km^2$. The full area of Turin is listed as $130km^2$, with extra-urban roads included.

[2] We will see these criteria in detail in the next section.

2.4 Possible Strategies

This type of structure allows for flexible experimentation with different traffic management strategies. It enables a direct comparison between a isolated control strategy[3] and a distributed communication-based coordination. In fact we are able to change and tune parameters to assess the response of the model in various traffic scenarios, changing vehicle density and agents' properties.

Algorithm 1 Graph Initialization

```
Require: Uploading Files
    file shape_file_roads ← roads file path;
    file shape_file_nodes ← nodes file path;
    geometry ← envelope(shape_file_roads)
Require: Building the graph
  graph the_graph;
  graph start_graph;
Ensure: Setting the driving property
  the_graph ← as_driving_graph(road, node)
  with_weights weight_map
```

3 Code Implementation

The implementation of the model is structured into three main components:

- **Map initialization**: here we generate the spatial layout of the map, based on real world GIS data obtained from OpenStreetMap[4].
- **Agents' Structure**: in this section we define, initialize and implement all agents, and the set of their possible actions.
- **Experiment Section**: we define multiple setup configurations to run and monitor our simulation, in order to simplify testing, performance evaluation, and parameters visualization.

3.1 Map Initialization

The simulation begins by building a graph-based representation of a selected area of the city. Here, roads are modeled as weighted and directed edges and intersections as nodes. The spatial layout generation is provided by QGIS software [5], which allows us to select the map we want through *OpenStreetMap* and then clean the data to generate the graph (for more details, see Appendix A). Once the graph is created, each node is initialized using the built-in *road_node* skill provided by GAMA[6]. Then road segments are classified according to our ordering algorithm (see Appendix B).

3.2 Agents' Structure

3.2.1 Vehicles

Once the road network is initialized, vehicle agents are introduced into the simulation. We defined two type of vehicles: **cars** and **buses**, each with distinct route properties. Car agents are assigned a randomly selected origin and destination from the list of available nodes in the graph. After a car has reached its destination, it is removed from the simulation. This behavior is typical of private vehicle trips in an urban environment, where the destination often changes and it is not repeated.

In contrast, bus agents follow predefined routes that correspond to *bus lines*. These lines are randomly generated at the start of each simulation run, by selecting a sequence of roads and intersections to form a path. Each bus chooses one of the available routes and follows it until the end of the run, simulating regular public transport with fixed schedule.

A key functional difference between the two vehicles, lies in their interaction with intersections. Bus agents actively communicate with upcoming traffic lights to request priority when approaching an intersection. This mechanism is intended to replicate real-world traffic conditions where buses are often given preferential treatment, such as dedicated lanes or priority signaling at intersections. On the other hand, cars just follow right-of-way rules and wait patiently in queues, following traffic lights' indications.

GAMA's built-in *driving skill* establish general behavior for vehicle movement. In particular it provides the default sequence of actions to follow, enforcing traffic rules to reflect driver behavior. To further enhance realism, we introduced a feature that mimics dedicated turn lanes. With a *switch lane* mode, we account for a *bias* that allows vehicles to put themselves in the left lane for turning, capturing the variability often seen in real traffic scenarios.

3.2.2 Traffic-Lights Agents

Traffic-light agents are the core decision-makers at intersections and can operate under three distinct control strategies: fixed timing, responsive, and communication-based flow coordination. Each con-

[3] Here we both refer to the timer-based inefficient scenario and the reactive one without communication. The latter is isolated in this sense.

[4] The website is `https://www.openstreetmap.org/`

[5] See the website `https://qgis.org/`

[6] See the documentation here: `https://gama-platform.org/wiki/BuiltInSkills`

Algorithm 2 General Vehicle Settings

Require: Species Declaration

```
species vehicle skills:  [driving] {
rgb color ← rnd_color(255);
float offset_distance ← 0.2;
init{
vehicle_lenght ← 3.8m;
max_speed ← 150km/h;
prob_resp_prior ← 0.75 + rnd(0.24);
prob_lane_up ← 0.2;
prob_lane_down ← 0.2;
prob_block_node ← 0.001; } }
```

Ensure: Movement when not done

```
reflex move when: final_target != nil {
    do drive; }
```

figuration progressively increases in complexity and adaptability, allowing us to analyze their impact on traffic flow and network performance.

- Fixed timing - This strategy replicates the behavior of traditional traffic lights in real-world systems. Each intersection follows a predefined cycle, switching between red and green states based on a fixed timer. To avoid perfect synchronization across the entire network - which would lead to artificial periodic congestion and non natural traffic patterns - a small random variation is added to the cycle duration for each intersection. This introduces small changes in period that reflect the variability observed in actual urban settings.

- Responsive - In the responsive strategy, traffic-light agents incorporate basic environmental awareness into their decision-making. Each intersection monitors the number of vehicles currently on its incoming roads. Based on this input, the traffic light can either maintain its current signal phase - following the default timing - or switch states earlier than scheduled if a high vehicle count is detected on a particular road. This hybrid approach introduces a moderate level of dynamics without requiring communication between agents. It acts as a transitional control scheme between the purely reactive fixed strategy and the fully cooperative communication-based system, helping to isolate the effects of local traffic awareness on system performance. It also enables us to identify which metrics are affected by each working mode.

- Communication-Based Flow Coordination - This advanced strategy implements decentralized coordination among traffic-light agents through structured message exchange, using the FIPA communication protocol. In this configuration, each intersection evaluates its state based on three main indicators:

- the number of vehicles currently queued on each of the incoming roads;
- the presence of priority vehicles, such as buses;
- incoming requests from adjacent intersections, which indicate the need to release congestion to guarantee a higher flow.

The request system is the key innovation of our system. A traffic-light agent will issue a request to a neighboring intersection if both of the following conditions are met:

1. vehicles count on its (the sender) incoming roads exceeds a predefined threshold;
2. at least one of its outgoing roads (which feeds into the neighboring intersection) is congested or blocked.

When these conditions are satisfied, the agent sends a request to the intersection at the downstream end of the jammed road, asking for a green phase on that segment. This mechanism enables cooperative decongestion, as traffic lights support each other in managing local bottlenecks and improving global flow efficiency. Final decisions about whether to switch the signal state or not, are based on a weighted evaluation of all active factors. To ensure that all possible directions are served fairly, we define both a minimum and a maximum duration for each green light phase. This prevents situations where less busy roads are forced to wait too long, or where a green light changes too quickly before vehicles can pass.

3.3 Experiments

Here we present how we explore the responses of the model to particular parameter combinations. The analysis will be presented in the next section. Our plan consists of the following steps:

- Validate the model. We choose to use the *mean traffic flux*, which is computed hourly, and the *fixed timing* traffic light configuration, considering only primary and secondary roads to compare our model with actual traffic data in Turin;
- Collect data with the 3 traffic-light configurations available. We define the variables of interest for our analysis, such as (positive) normalized cumulative acceleration, mean vehicles speed, spatial delta accounting for reroutes, and stress tests focusing on rush hours;
- We analyze the data and gather results and conclusions, and eventually we suggest future developments and research topics.

.

4 Execution & Analysis

In this section we present the parameters choice and the procedures we followed, the model validation, and data analysis.

4.1 Parameters Initialization

As we can see in Alg.1, we need to upload roads and nodes files, in order to create the network. Then we build the graph and we include the driving property.

Agents initialization is a fundamental step in our model: in Alg.2 we see the general settings. We create the species *vehicle* with fundamental characteristics such as size and speed, and probabilities of respecting priorities and changing lanes (if the road has more than one lane). These values are taken from GAMA example models [7]. It is important to note that this is just the general initialization for the species, then we specify children properties accordingly (cars and buses). We used data from the city of Turin Website [8] to initialize the correct number of vehicles in our network. We scaled it accordingly to our map and hours during the day (peak hour is about 18000 vehicles in all the city, which is scaled to roughly 5000 for our map).

Algorithm 3 Fixed Timing Traffic Lights

Require: Reflex and Timer
  `reflex update_state when traffic_light:`{
  `if fixed_timer:`{
  `timer` $\leftarrow$ `timer +1`;
  `if timer` $\geq$ `max_timer` : {
  `do switch_state`}}}

Ensure: Switching action
  `action switch_state`{
  `stop` $\leftarrow$ `road_e_ok ?roads_in_e :roads_in_o`;
  `road_even_ok` $\leftarrow$ `!roads_even_ok`;
  `color` $\leftarrow$ `#red ? #green : #red`;
  `timer` $\leftarrow$ 0}

Then we initialize traffic lights configuration[7]. During the simulation we will be able to switch it in case of need. For this reason we need a *reflex* to update the state and a series of cases nested in it. As we can see in Alg.3, the fixed timing configuration is pretty simple: if the timer exceeds the maximum, the light switches state, replicating real world intersections.

[7]Here we propose only the fixed timing mode. For more details: `https://github.com/marcobianchi463/AB-lights`

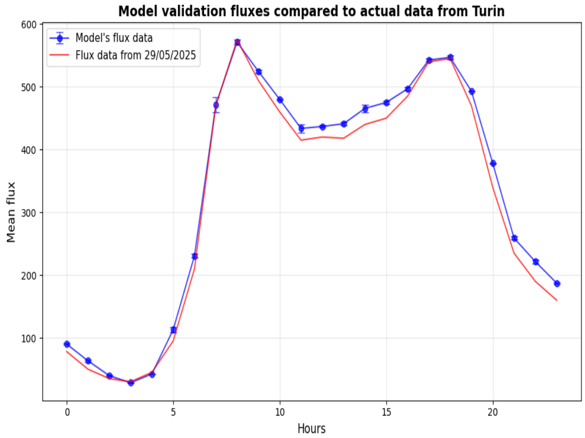

Figure 1: Hourly mean flux of vehicles. Our simulation (blue) is compared with real data (red) collected on 29/05/2025

4.2 Model Validation

To validate the model, we used public available data given by the city of Turin [8]. We followed their approach, computing the hourly mean flux of vehicle in principal roads. We set the *fixed timing* mode and we collected data from multiple runs with different seeds. This was done to mimic the collection of data in different initial conditions, and then we computed the mean value and the standard deviations. A comparison with real data is plotted in Fig.1[8]. As we can see there is a nice overlapping, giving us enough reasons to establish the validation of our model.

4.3 Data Analysis

Our main goal is to evaluate the effect of traffic lights logic on traffic flow efficiency, using the positive cumulative acceleration, normalized to the total number of trips, as a proxy for fuel consumption and emissions. A reduction in this metric would indicate a gain in environmental and operational efficiency. Other indicators include mean vehicle speed, a spatial delta accounting for path deviation, and the percentage of stopped vehicles. As a robustness check we also consider the hourly mean flux of vehicles. Before starting the analysis we expected a decrease in the spatial delta and stopped vehicles, and an increase in speed and hourly flux.

4.3.1 Traffic lights logic comparison

To compare the strategies in plots, we ran a simulation for each configuration with the same seed. Then to have some statistical measures, we ran a set of simulations with a group of seeds, took the mean

[8]As no information on the data errors is provided from the *Simone* project, we assumed them without error

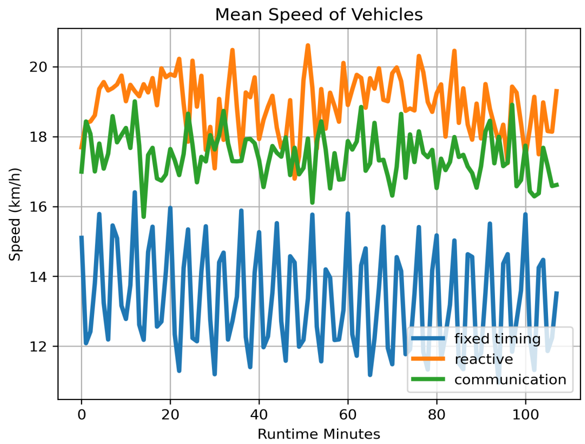

Figure 2: Comparisons of mean speed for the three traffic lights configurations.

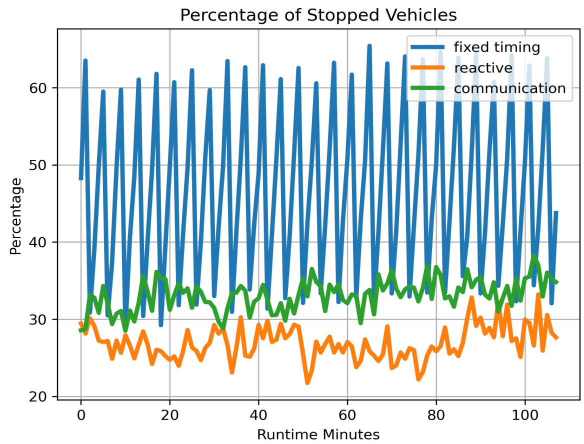

Figure 3: Comparison between the percentages of stopped vehicles for different traffic lights configuration.

and propagated errors[9] and with a medium network load of 3500 vehicles, allowing us to average results and estimate statistical uncertainty. Fig.2 shows the trend in mean vehicle speed across the three configurations. A substantial improvement is observed when switching from fixed to reactive logic: the average speed increases from 13.83 ± 0.15 km/h to 19.32 ± 0.10 km/h, representing an increase of approximately 30%. Surprisingly, the communicative configuration does not outperform the reactive one under these conditions, reaching a slightly lower mean value of 17.95 ± 0.07 km/h. While the communication logic is expected to mitigate traffic jams, in this particular setup it seems to introduce slight inefficiencies. This might be due to the communication overhead or the lack of severe congestion scenarios where coordination would shine.

Further insights are provided by analyzing the percentage of stopped vehicles and the normalized acceleration, as reported in Tables [1] and [2], Fig.[3] and Fig.[4]). Here, both the reactive and communicative strategies reduce the percentage of vehicles at near-zero speed, suggesting improved flow stability. Likewise, cumulative acceleration decreases significantly with adaptive control logic. Moreover, communicative and reactive strategies yield comparable values, indicating similar levels of smoothness in traffic flow. Finally we report the hourly flux of vehicles in Table [3], which reflects the above considerations.

[9] Here we used five seeds to mimic a real analysis made on different days. Then we made fits to have mean values and errors, as reported in tables.

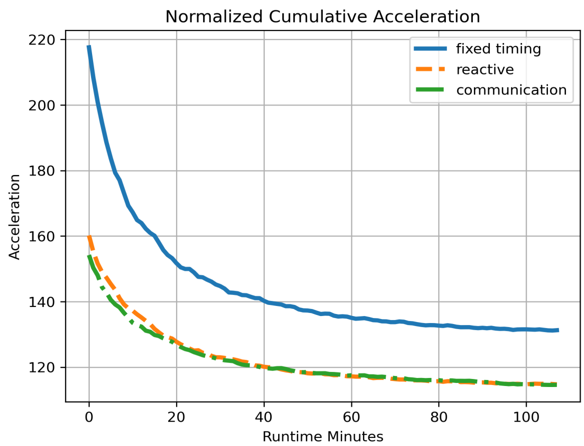

Figure 4: Comparison between cumulative accelerations, normalized by the number of trips completed.

Configuration	% of Stopped Vehicles
Fixed Timing	45.0 ± 1.2
Reactive	25.5 ± 0.2
Communication	30.8 ± 0.2

Table 1: Mean percentage of stopped vehicles across five simulations with different seeds

Configuration	Normalized Acceleration
Fixed Timing	143.61 ± 0.14
Reactive	122.30 ± 0.08
Communication	122.57 ± 0.09

Table 2: Mean measurement of cumulative acceleration normalized on total number of trips

Configuration	Hourly Flux of Vehicles
Fixed Timing	440 ± 4
Reactive	619 ± 4
Communication	575 ± 6

Table 3: Hourly Flux of Vehicles mediated across five different seeds.

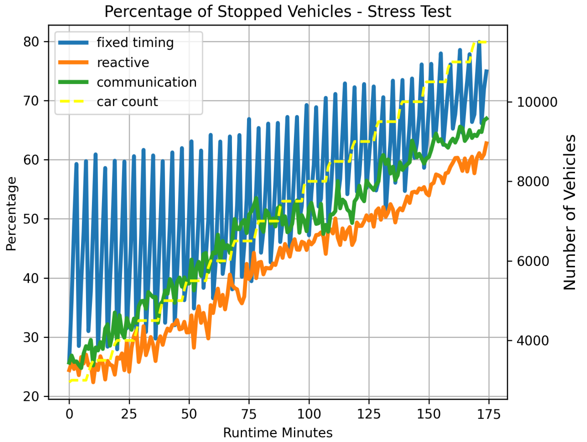

Figure 5: Percentage of stopped vehicles during a rapid increase in car counts

4.3.2 Saturation Test and Model Robustness

To evaluate how the model behaves under increasing traffic pressure, we conducted a stress test. In this scenario, we progressively increased the number of vehicles from 2000 to 12000, in batches of 500 every 10 minutes, using the same seed but different traffic light strategies. The goal was to observe how each configuration manages growing congestion, way above peak hour limit.

As shown in Figures [5] and [6], both reactive and communicative configurations outperform the fixed-timing strategy in maintaining lower congestion and higher average speeds as load increases.

Interestingly, the mean delta —a measure of how far vehicles deviate from the optimal path— shows a curious behavior. While reactive lights cause the delta to increase "linearly", the communicative configuration maintains a lower and more stable deviation. This suggests that coordinated decision-making via communication helps avoid unnecessary rerouting caused by local congestion, effectively reducing blockage propagation across the network.

Additionally, the communicative strategy introduces a temporary stabilization in both stopped vehicle and average speeds slope at high traffic volumes. While performance eventually degrades under extreme load—as expected—the coordinated system delays the rate of saturation. This "aftershock mitigation" demonstrates the potential of agent communication in postponing systemic breakdowns in traffic flow.

5 Results

Our simulations demonstrate how incorporating communication among traffic lights can improve the

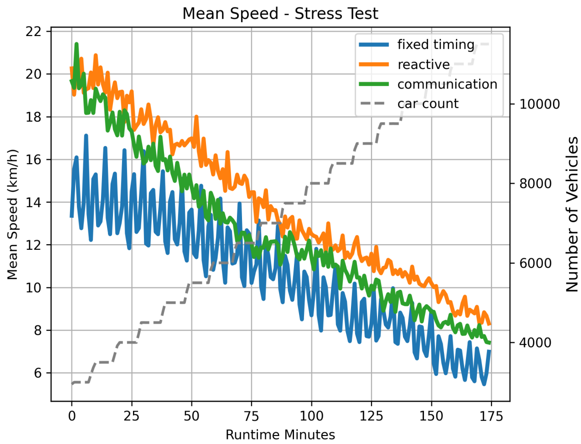

Figure 6: Comparisons of mean velocities for the three traffic lights configurations

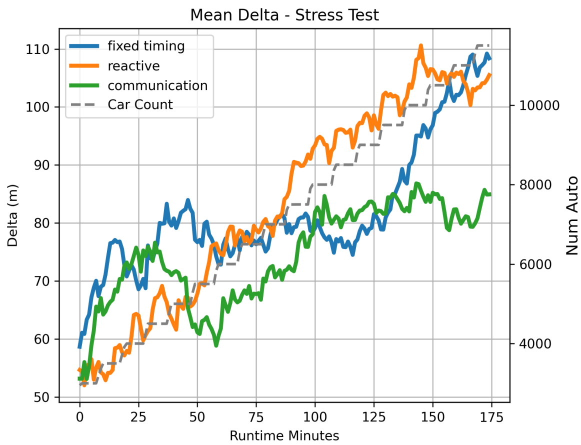

Figure 7: Path length deviation from shortest, mediated over all vehicles

overall performance of an urban traffic system under certain conditions. Specifically, switching from fixed-time to adaptive strategies leads to measurable improvements in key metrics, including:

- Increased mean vehicle speed and hourly flux;
- Reduced positive cumulative acceleration, a proxy for lower fuel consumption and emissions;
- Greater resilience to congestion, particularly during periods of high vehicle inflow;

One of the core innovations of this study lies in the comparison between purely reactive and locally coordinated (communicative) traffic lights using the FIPA-ACL protocol. This coordination enables intersections to request green-light priority to prevent local congestions.

However, the data show that purely reactive traffic lights slightly outperform the communicative configuration in normal traffic conditions. This suggests that communication introduces some overhead or latency that does not translate into measurable gains when traffic loads are moderate and homogeneous.

Conversely, communicative traffic lights exhibit improved behavior during stress scenarios, where traffic density increases rapidly. In these conditions, coordination allows for more effective mitigation of localized congestion and enables vehicles to maintain shorter path lengths by reducing the number of blocked intersections. Thus, communication appears to add value primarily in saturated environments, rather than under everyday conditions.

To summarize, we identify clear advantages of adaptive methods in improving traffic flow, reducing emissions, and increasing resilience to congestion. While purely reactive logic performs best under normal traffic loads, communicative traffic lights show promising benefits in high-density scenarios. These findings support the potential of targeted coordination and adaptive infrastructure in building smarter, more efficient urban mobility systems.

In conclusion, we suggest several directions for future research to address the limitations of our model:

- Apply communicative logic only to key intersections, such as high-volume arteries or bottlenecks, while using reactive logic elsewhere. This could preserve the resilience benefits while minimizing system complexity;
- Introduction of more agents such as pedestrians and bicycles, and more detailed bus behavior, especially in relation to signal priority and shared spaces;
- Further explore the use of acceleration and braking profiles to build energy-efficient traffic control logic, aligning urban mobility with sustainability goals.

A QGIS map elaboration

To generate the spatial environment for our simulation, we selected a portion of the urban area of Turin and obtained its geographic data from OpenStreetMap (OSM). These data were processed using QGIS [2] to convert the map into a .GIS file format, suitable for use in GAMA. However, this conversion is challenging and requires a careful cleanup phase to ensure an accurate and functional structure within the simulation framework.

One issue encountered in OSM data is the presence of imprecise or overlapping road segments, particularly where roads should intersect multiple times but are instead misaligned. This results in broken connectivity within the graph and prevents vehicles from traveling across adjacent routes. Additionally, non-vehicular paths such as bike lanes and walkways must be removed to avoid interference with traffic simulation logic.

A further critical step involves flattening the map from a 3D real-world layout to a 2D graph. While necessary for simplifying computations, this introduces problems in areas with elevation such as overpasses and underpasses. Once flattened, these structures may appear as standard intersections in the 2D graph, creating overlapping and false connections between roads that do not actually meet in reality. These inaccuracies can severely disrupt vehicle routing, often leading to artificial congestion on certain roads while adjacent (but disconnected) roads are not occupied.

To resolve these issues, we performed manual corrections using QGIS tools. As said above, this involved adjusting problematic intersections, aligning road segments, and redefining node connectivity where automated procedures failed. Though time-consuming, this step was essential for ensuring that the simulation environment faithfully represents a traversable and realistic urban network.

B Roads ordering algorithm

A critical challenge during map initialization was the ordering of roads connected to each intersection node. When the graph is built from the GIS data, each intersection receives a list of connected roads.

However, these roads are listed in an unreasonable order—typically based on how they are stored or indexed in the source files—making it impossible to consistently determine which roads should be assigned to the even or odd set used by the traffic-light system.

To address this, we developed a custom ordering algorithm that runs during the initialization phase. For each intersection, the algorithm reads the geometric orientation (angle) of all connected roads and orders them counterclockwise starting from a fixed reference direction (e.g., due north). Once ordered, the roads are systematically divided into even and odd subsets. These groups are essential for managing intersection behavior: they provide a structured way to model right-of-way rules, reduce ambiguity in road referencing, and support decision-making by traffic-light agents. For instance, this classification simplifies the handling of T-junctions or irregular intersections, where fewer than four roads are present. Additionally, this system helps identify false intersections that may result from inconsistencies in GIS data—such as when a single continuous road is split into segments with different names. In such cases, intersections may be incorrectly generated, and the algorithm helps flag or reclassify them based on the geometric arrangement and connectivity.

This orientation-based ordering mechanism is a core component of the model's logic and contributes to both traffic-light control accuracy and realistic vehicle navigation throughout the whole simulation.

Contents

References

[1] Raul Cajias, Antonio Gonzalez-Pardo, and David Camacho. A multi-agent traffic simulation framework for evaluating the impact of traffic lights. In *International Conference on Agents and Artificial Intelligence (ICCART)At: Rome, ItalyVolume: SciTePress – Science and Technology Publications. pp. 443-446*, pages 443–446, 01 2011.

[2] QGIS Foundation. Qgis platform. `https://qgis.org/`, 2016.

[3] GAMA. Gama platform website. `http://gama-platform.org`, 2025.

[4] Giancarlo Genta. *Meccanica dell'Autoveicolo*. Levrotto & Bella, 5 edition, February 2000.

[5] Gerhard Lechner, Harald Naunheimer, and Joachim Ryborz. *Automotive Transmissions: Fundamentals, Selection, Design, and Application*. Springer, Berlin, 1999.

[6] Marcel Rieser. Modeling public transport with matsim. In Andreas Horni, Kai Nagel, and Kay Axhausen, editors, *The Multi-Agent Transport Simulation MATSim*, pages 85–98. Ubiquity Press, 2016.

[7] Patrick Taillandier. Traffic simulation with the gama platform. In *Proceedings of the Eighth International Workshop on Agents in Traffic and Transportation (ATT)*, 05 2014.

[8] Muoversi A Torino. Città di torino (website). `http://www.muoversiatorino.it/it/numeri-mobilita/`, 2025.

[9] Allan Vital, Vitor Navarro, Fabricio Barth, and Orlando Rodrigues Jr. Development of intelligent traffic lights using multi-agent systems. In *Proceedings Media of the 12th World Conference on Transport Research*, pages 1–24, July 2010.

[10] Ali Wided. Traffic management system and traffic light control in smart city to reduce traffic congestion. *International Journal of Automation and Smart Technology*, 07 2023.